Microelectronics Journal 41 (2010) 247–255

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

Standby power consumption estimation by interacting leakage current mechanisms in nanoscaled CMOS digital circuits

Paulo F. Butzen [a,*], Leomar S. da Rosa Jr[b], Erasmo J.D. Chiappetta Filho [a], André I. Reis [a], Renato P. Ribas [a]

[a] Federal University of Rio Grande do Sul, Av. Bento Gonçalves, 9500, Bloco IV, CEP 91501-970, Porto Alegre, RS, Brazil
[b] Federal University of Pelotas, Rua Gomes Carneiro, 1, CEP 96010-610, Pelotas, RS, Brazil

ARTICLE INFO

Article history:
Received 14 May 2009
Received in revised form
25 February 2010
Accepted 1 March 2010
Available online 21 March 2010

Keywords:
Static power dissipation
Subthreshold leakage
Gate oxide tunneling
CMOS complex gates

ABSTRACT

Leakage currents are gaining importance as design parameters in nanometer CMOS technologies. A novel leakage current estimation method, which takes into account the dependency of leakage mechanisms, is proposed for general CMOS complex gates, including non-series–parallel transistor arrangements, not covered by existing approaches. The main contribution of this work is a fast, accurate, and systematic procedure to determine the potentials at transistor network nodes for calculating standby static currents. The proposed method has been validated through electrical simulations, showing an error smaller than 7% and an 80× speed-up when comparing to electrical simulation.

1. Introduction

Leakage currents are one of the major design concerns in deep submicron technologies due to the aggressive scaling of MOS device [1,2]. Supply voltage (V_{dd}) has been reduced to keep power consumption under control. As a consequence, the transistor threshold voltage (V_{th}) is also scaled down to maintain the drive current capability and to achieve performance improvement when reducing the technology node. However, the threshold voltage reduction increases subthreshold current exponentially. Moreover, short channel effects (SCE), such as drain-induced barrier lowering (DIBL), are being reinforced when the technology shrinking occurs. Hence, oxide thickness (t_{ox}) has to follow this reduction but at the expense of significant gate tunneling leakage current. High-κ dielectrics are being considered as a promising way to control gate leakage [3], but even for high-κ it is not possible to neglect the gate leakage interaction with other components. Furthermore, higher doping profile results in increasing reverse-biased junction band-to-band tunneling (BTBT) current, although it is expected to become relevant only for technologies sub-25 nm [4].

Subthreshold leakage current occurs in off-devices (transistors which are turned off), presenting relevant values in transistor with channel length shorter than 180 nm [5]. In terms of subthreshold leakage saving techniques and estimation models, the *stack effect* represents the main factor to be taken into account [6–9]. Gate oxide leakage, in turn, is verified in both on- and off-transistor when different potentials are applied between drain/source and gate terminals [10]. In sub-100 nm processes, where gate oxide thickness is smaller than 16 Å, subthreshold and gate oxide leakages tend to present the same order of magnitude [11]. As a result, the interaction among them in the total circuit leakage computation should not be neglected.

This paper presents a new analytical method for total leakage power estimation in CMOS logic gates, considering the interaction of both subthreshold and gate oxide leakage mechanisms. The paper is organized as follows. The motivation of this work is pointed out in Section 2. Section 3 presents the leakage current equations and the process for extraction of parameters applied in the method. The leakage estimation procedure is then described in Section 4. Experimental results, considering single gates and benchmark circuits, are presented in Section 5. Finally, the conclusions are outlined in Section 6.

2. Motivation

Analytical leakage estimators are quite useful to speed-up the standby power consumption prediction in CMOS digital circuits, once numerical simulations are time consuming even for such a DC analysis. Several methods have been presented in the literature for CMOS leakage estimation at gate level [5–10,12–16]. Subthreshold current alone is well analyzed in [5–9]. These works treat only

* Corresponding author: Tel.: +55 51 3308 6165; fax: +55 51 3308 7308.
E-mail addresses: pbutzen@inf.ufrgs.br (P.F. Butzen), leomarjr@ufpel.edu.br (L.S. da Rosa Jr), ejdcfilho@inf.ufrgs.br (E.J. Chiappetta Filho), andreis@inf.ufrgs.br (A.I. Reis), rpribas@inf.ufrgs.br (R.P. Ribas).

0026-2692/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mejo.2010.03.003

single transistor stacks, while parallel devices in the electrical arrangement are primarily converted to equivalent single devices. On the other hand, gate leakage current alone is predicted in [10] by considering transistor biasing and full voltage swing at internal nodes, compromising the accuracy of the method.

The interaction between subthreshold and gate leakage currents is evaluated in [12–16]. In [12,13], the components are pre-characterized through electrical simulations according to the transistor biasing. The polarization patterns used in the pre-characterization are identified in the logic gate topology under a certain steady state condition, and the correspondent leakage value is then included in the total cell leakage calculation. In [14], Hertani et al. uses similar strategy of pre-extracted data to obtain the subthreshold leakage in off-device stack arrangements. Such information is used to assign the internal node voltages in order to predict the gate oxide current for the total leakage computation. Yang et al. [15] uses similar methods proposed separately for subthreshold [6] and gate oxide [10] leakages to estimate the total static current by summing these partials. The actual leakage mechanisms interaction at internal gate nodes is ignored. Moreover, notice that the estimation accuracy presented in [12,15] has been demonstrated for the case when the gate current value is significantly higher than the subthreshold one. Otherwise, some assumptions in those works are no longer valid and the results are compromised. In [16], an analysis considering detailed leakage mechanisms interaction is presented by Mukhopadhyay et al. MOS transistors are modeled as a combination of voltage controlled current sources (SCS), and a numerical solver is used to estimate leakage in logic gates by solving the Kirchoffs Current Law (KCL) system at intermediate nodes.

In summary, most of the previous related works have their applicability restricted to simple logic gates, with exception of [16] where the complexity of the method and the computation time are the main drawbacks. Hence, CMOS complex gates, as the ones that implement the GenLib_44-6 library functions [17], and non-series–parallel arrangements, like the ones proposed by Kagaris et al. [18], cannot be treated by existing approaches. Examples of such complex gates are depicted in Fig. 1.

Indeed, an effective modeling of the interaction between different leakage mechanisms for the total standby current calculation is necessary, since they affect the internal node voltages in the logic gates, and the leakage currents are directly related to those potentials. An example frequently found in the literature to demonstrate leakage prediction methods is illustrated in Fig. 2 [12–14]. On that, there is a conducting

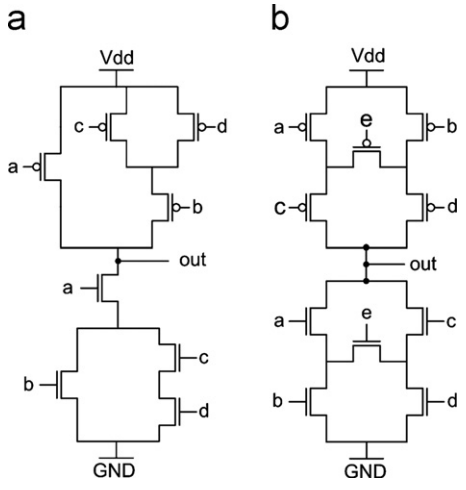

Fig. 1. CMOS complex gates: (a) out=!(*a*.(*b*+*c*.*d*)), and (b) out=!((*a*.(*b*+*d*.*e*))+(*c*.(*d*+*b*.*e*))).

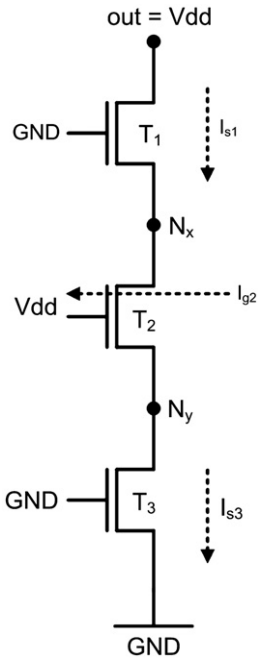

Fig. 2. Leakage currents in a three transistors stack.

Table 1
Total leakage current in the transistor stack depicted in Fig. 2, for 32 nm CMOS PTM process [19]: bulk and high-K options.

32 nm CMOS PTM	Bulk process (nA)	High-K process (nA)
I_s	19.76	7.09
I_s+I_g	45.48	9.23
I_s and I_g	26.94	7.85

NMOS transistor T_2 between two off-devices T_1 and T_3. In this case, the nodes N_x and N_y assume the same potential due to the on-transistor T_2. This potential is then found by computing the subthreshold currents (I_{s1} and I_{s3}) and the gate current I_{g2}, increasing so the voltage value when compared to the analysis considering only subthreshold currents. Table 1 presents the total leakage current values obtained for three different situations: taking into account only subthreshold component (I_s), computing separately subthreshold and gate oxide leakages (I_s+I_g), and considering the leakage mechanisms interaction (I_s and I_g). It clearly shows that the leakage interaction must not be neglected in the total leakage analysis even for high-K CMOS process.

Moreover, although several authors consider the on-transistors in the off-plane of a CMOS gate as ideal short-circuit [6–10], this is only acceptable when such conducting device is not connected directly to the output node [12–14,16]. Otherwise, the voltage drop across the on-device is quite significant for leakage estimation, being approximately the nominal threshold voltage. The leakage currents indicated in Fig. 3 illustrate this effect using 32 nm bulk CMOS PTM parameters [19] and temperature at 80 °C. The resulting currents are: I_{g1}=5.8 nA and I_{g2}=1.3 nA; I_{s1}=768.8 nA and I_{s2}=222.6 nA.

The major contribution of this work is a simple, fast, and accurate analytical method for total leakage power estimation in general CMOS complex gates, considering both subthreshold and gate oxide leakage components. The interaction between these two leakage mechanisms occurs during the determination of internal node voltages at gate level. A systematic procedure has been defined to determine the node potentials instead of applying

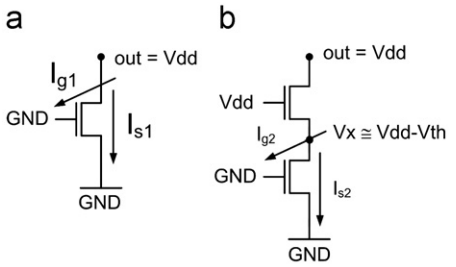

Fig. 3. Leakage currents in two different off-transistor conditions: (a) single off-device and (b) two stacked transistors.

a more time consuming numerical solver, without penalty in accuracy.

3. Leakage mechanisms

For nanometer MOS devices, leakage current is dominated by subthreshold and gate oxide tunneling currents. There are other leakage components, like band-to-band tunneling (BTBT) leakage, expected to become relevant in process sub-25 nm, gate induced drain leakage (GIDL) and punchthrough current, which can be neglected in normal operation mode [4]. This section describes the formulation used to model subthreshold and gate currents, as well as the extraction of technology parameters required in the proposed equations.

3.1. Subthreshold current

As devices are shrinking, supply voltages have been scaled down to keep dynamic power consumption in acceptable levels. However, in order to maintain drive current capability and short channel effects (SCE) under control, the threshold voltage has also been reduced, but at penalty of increasing subthreshold current. Such leakage component means a current flowing between drain and source terminals when the device is operating in a weak inversion condition. Thus, this leakage current depends on the transistor threshold voltage, gate-to-source and drain-to-source voltages, and operating temperature.

From BSIM MOS transistor model [20], the subthreshold current for a MOSFET device can be expressed as:

$$I_s = I_0 e^{(V_{gs}-V_{th}/nV_T)}\left[1-e^{-(V_{ds}/V_T)}\right] \quad (1)$$

being $I_0 = (W\mu_0 C_{ox} V_T^2 e^{1.8})/L$, V_T is the thermal voltage, V_{th} is the threshold voltage, V_{ds} is the drain-to-source voltage and V_{gs} is the gate-to-source voltage. W and L are the effective transistor width and length, respectively. C_{ox} is the gate oxide capacitance, μ_0 is the carrier mobility, and n is the subthreshold swing coefficient.

The threshold voltage, in turn, can be expressed by:

$$V_{th} = V_{t0} - \eta V_{ds} - \gamma V_{bs} \quad (2)$$

where V_{t0} is the zero bias threshold voltage, V_{bs} is the bulk-to-source voltage, η is the DIBL coefficient, and γ is the body effect coefficient that can be linearized for small values of V_{bs}.

Since the proposed method uses the internal potentials at the logic gate topology (transistors arrangement) to estimate the cell leakage, Eq. (1) has been re-written in order to represent the subthreshold current in terms of these circuit voltages (V_{gs}, V_{ds}, and V_{bs}), transistor sizes (W and L), and fixed parameters (empirical constants), as following:

$$I_s = I_{s0}\frac{W}{L} e^{(V_{gs}+\eta V_{ds}+\gamma V_{bs}/nV_T)}\left[1-e^{-(V_{ds}/V_T)}\right] \quad (3)$$

where $I_{s0} = \mu_0 C_{ox} V_T^2 e^{1.8V_{t0}/nV_T}$. Considering $n \approx 1.45$ for nanometer technologies [6], the remaining parameters of the equation (I_{s0}, η, and γ) are extracted from electrical simulations, as described later.

3.2. Gate oxide tunneling current

As mentioned before, aggressive device scaling in nanometer regime increases SCE. In order to maintain a reasonable immunity to such effects, the oxide thickness must also become thinner at each new technology node. However, it gives rise to high electric field, resulting in direct tunneling current through transistor gate insulator. This gate leakage current depends on the potential across the oxide (V_{ox}) and the oxide thickness (t_{ox}). It can be modeled by [1]:

$$I_g = WLA\left(\frac{V_{ox}}{t_{ox}}\right)^2 \exp\left(\frac{-B(1-(1-(V_{ox}/\phi_{ox}))^{3/2})}{V_{ox}/t_{ox}}\right) \quad (4)$$

being $A = q^3/16\pi^2 h\phi_{ox}$ and $B = 4\pi\sqrt{2m_{ox}}\phi_{ox}^{3/2}/3hq$. m_{ox} is the effective mass of the tunneling particle, ϕ_{ox} is the tunneling barrier height, h is $1/2\pi$ times Planck's constant, and q is the electron charge.

Similar to Eq. (3), as the proposed estimation method explores the transistor network internal voltages to calculate the leakage values, the gate current formulation can be re-written in a simplified way, emphasizing the gate voltage (V_{gs}) and transistor geometry dependence, as follows:

$$I_g = I_{g0} WL e^{-(K/|V_{gs}|)} \quad (5)$$

where I_{g0} is the gate leakage current for V_{gs} equal to V_{dd}, and K is a calibration constant, based on the difference between gate leakage currents when $V_{gs}=V_{dd}$ and $V_{gs}=0.9V_{dd}$. Such empirical formulation presents good correlation to HSPICE simulation data, as demonstrated in Fig. 4, for 32 nm bulk CMOS PTM technology [19].

3.3. Extraction of parameters

The parameters I_{s0}, η, and γ as well as I_{g0} and K, required in Eqs. (3) and (5), are extracted from electrical simulations taking into account single MOS device. The circuit used in the simulations to extract the parameters is illustrated in Fig. 5. The simulation and extraction procedure has to be done only once for each technology process before applying the estimation method.

K, η, and γ constants are obtained by taking into account four pre-simulated currents, according to the voltage sources biasing indicated in Table 2. The subthreshold reference (I_{s_REF}) and the evaluation (I_{s_EVAL}) currents are used to compute the η and γ constants, as described in Eqs. (6) and (7). The gate leakage reference (I_{g_REF}) and the evaluation (I_{g_EVAL}) currents are used to

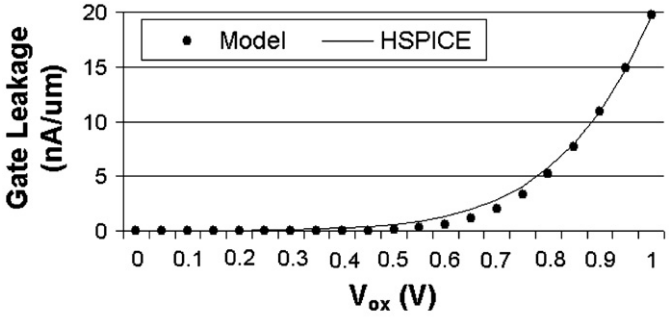

Fig. 4. Gate leakage current correlation of Eq. (5) to HSPICE data.

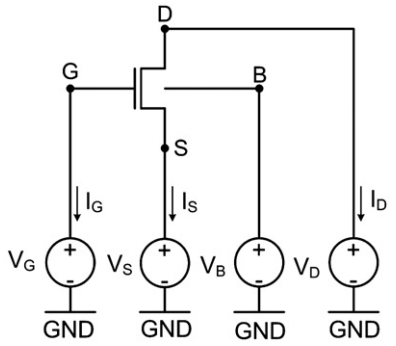

Fig. 5. Test structure for parameters extraction.

Table 2
Source biasing of test structure in Fig. 5 for parameters extraction.

Constant	Current	V_S	V_G	V_D	V_B
Kon_N	I_{g_REF}	GND	V_{dd}	GND	GND
Kon_N	I_{g_EVAL}	GND	$0.9{*}V_{dd}$	GND	GND
Koff_N	I_{g_REF}	GND	GND	V_{dd}	GND
Koff_N	I_{g_REF}	GND	GND	$0.9{*}V_{dd}$	GND
η*_N*	I_{s_REF}	GND	GND	V_{dd}	GND
η*_N*	I_{s_REF}	GND	GND	$0.9{*}V_{dd}$	GND
γ*_N*	I_{s_REF}	GND	GND	V_{dd}	GND
γ*_N*	I_{s_REF}	GND	GND	V_{dd}	$0.1{*}V_{dd}$
Kon_P	I_{g_REF}	V_{dd}	GND	V_{dd}	V_{dd}
Kon_P	I_{g_REF}	V_{dd}	$0.1{*}V_{dd}$	V_{dd}	V_{dd}
Koff_P	I_{g_REF}	V_{dd}	V_{dd}	GND	V_{dd}
Koff_P	I_{g_REF}	V_{dd}	V_{dd}	$0.1{*}V_{dd}$	V_{dd}
η*_P*	I_{s_REF}	V_{dd}	V_{dd}	GND	V_{dd}
η*_P*	I_{s_REF}	V_{dd}	V_{dd}	$0.1{*}V_{dd}$	V_{dd}
γ*_P*	I_{s_REF}	V_{dd}	V_{dd}	GND	V_{dd}
γ*_P*	I_{s_REF}	V_{dd}	V_{dd}	GND	$1.1{*}V_{dd}$

compute the K constants, as described in Eq. (8). These constants are then calculated as follows:

$$\eta = \frac{10nV_T \ln(I_{s\_REF}/I_{s\_EVAL})}{V_{dd}} \tag{6}$$

$$\gamma = \frac{-10nV_T \ln(I_{s\_REF}/I_{s\_EVAL})}{V_{dd}} \tag{7}$$

$$K = 10\ln(I_{g\_REF}/I_{g\_EVAL}) \tag{8}$$

Table 2 presents the voltage biasing for both transistor types, NMOS (_N) and PMOS (_P). Since the gate leakage occurs in conducting (on-) and non-conducting (off-) transistors, I_{g0} and K parameters must be extracted for both cases.

I_{s0} and I_{g0} parameters are then obtained from electrical simulation considering the previous extracted parameters and Eqs. (3) and (5), respectively.

4. Leakage estimation method

From previous sections, it is evident that the interaction between the leakage components is crucial to accurately estimate the total standby power dissipation in CMOS digital circuits.

In the scope of this work, a general CMOS logic gate corresponds to the static design style composed by two planes or networks, the pull-up PMOS plane, which connects the cell output node to power supply (V_{dd}), and the pull-down NMOS plane, which connects the output node to ground (GND) terminal, as shown in Fig. 1. 'On-plane' and 'off-plane' terms indicate the conducting and non-conducting planes, respectively, according to the steady state condition. From the off-plane is derived the 'off-network' that represents, in fact, the actual transistor arrangement responsible for isolating the output node voltage to the opposite power supply (V_{dd} or GND). The extraction of the off-network arrangement is important since the subthreshold leakage currents are observed only in the off-devices present in that plane.

The proposed algorithm is a unidirectional procedure, i.e., it does not present any feedback or loop. Algorithm steps are depicted in Fig. 6, and described in the following:

(1) Identification of the off-plane (pull-up or pull-down), according to the input logic value.
(2) Extraction of the off-network from the off-plane considering the on/off static condition of devices. On-devices are short-circuited and replaced by current sources, representing the gate leakage current contribution from such removed transistor. In this step, two situations may be identified and treated:
  (2.1) When on-device connects two internal nodes, parallel off-devices and transistor clusters (sub-networks) are eventually removed. In this case, each removed device from this sub-network must be replaced by a respective current source connected at this node, in order to keep the influence of its gate leakage contribution at the node potential, and to be considered in the total leakage calculation.
  (2.2) When on-device (in off-plane) is connected to the output node, it is removed but a differential voltage ΔV must be considered at the gate output voltage. It

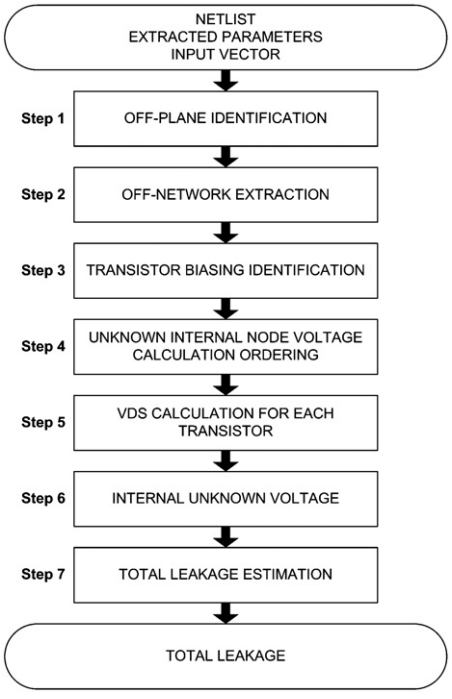

Fig. 6. Steps of leakage estimation method.

means, the output terminal assumes '$V_{dd}-\Delta V_N$' value instead of V_{dd} one in the case of the pull-down NMOS off-plane, and 'GND+ΔV_P' value instead of GND reference for the pull-up PMOS off-plane. This differential voltage ΔV can initially be considered as the transistor threshold voltage (V_{th}). However, this value is directly related to the transistor network topology. The tradeoff of the ΔV value is discussed in Section 5.3.1.

(3) Identification of DC polarity (biasing) of each off-device present in the off-network. It is a straightforward task for purely series–parallel transistor arrangements, by just evaluating the distance of transistor nodes to the off-network terminals. The 'distance' is defined here as the number of off-devices present in the shortest path to reach the network terminals (V_{dd} or GND, and output). In the case of non-series–parallel topologies, this task is not so obvious, as discussed in detail in Section 4.2.

(4) Ordering internal nodes in the off-network whose voltage value must be estimated. This is done according to the internal nodes distance to the output terminal. When different nodes present the same distance to the output, the distance to the supply terminal (V_{dd} or GND) is then taken into account, giving priority to the farthest node. In the case of nodes with equal distance, the ordering of the nodes is irrelevant and therefore it is chosen randomly.

(5) Calculation of drain-to-source voltage (V_{ds}) of each transistor by applying the Kirchhoff's Current Law (KCL) at each internal node. Differently from approaches that solve KCL system [16], the purpose here is to calculate KCL equations in a pre-defined order, according to precedent Step (4). To solve the equations, all unknown node voltages are initially considered at GND or V_{dd} potential, depending on the type of the off-plane, pull-down NMOS or pull-up PMOS, respectively. The loss in accuracy by using such strategy is not so relevant, as analyzed in Section 5.3.1.

(6) Definition of the voltage at each node based on the transistor V_{ds} voltages, previously calculated in Step (5). Starting from V_{dd} or GND terminal, the potential at each unknown node is computed by summing the V_{ds} values of each transistor in the path 'node-supply', respecting the inverse order established in step (4). When a given node has more than one possible voltage value, i.e. there is more than one path to reach the supply terminal, the voltage at this node is determined by the highest value obtained in the case of a NMOS off-network and by the lowest value in a PMOS network. The impact of using the highest and the lowest values according to the network type is discussed in Section 5.3.1.

(7) Estimation of the total leakage current considering the internal node voltages, previously determined in Step (6). It corresponds to the sum of all leakage currents flowing from V_{dd} terminal or to GND terminal. For instance, considering the currents flowing to GND, the total leakage is given by the sum of the subthreshold current of all transistors directly connected to that terminal, the gate current through on- and off-devices in the off-plane, and the gate current through on-devices in the on-plane.

4.1. Case study: method demonstration

To illustrate the procedure previously described, consider the complex gate depicted in Fig. 7a.

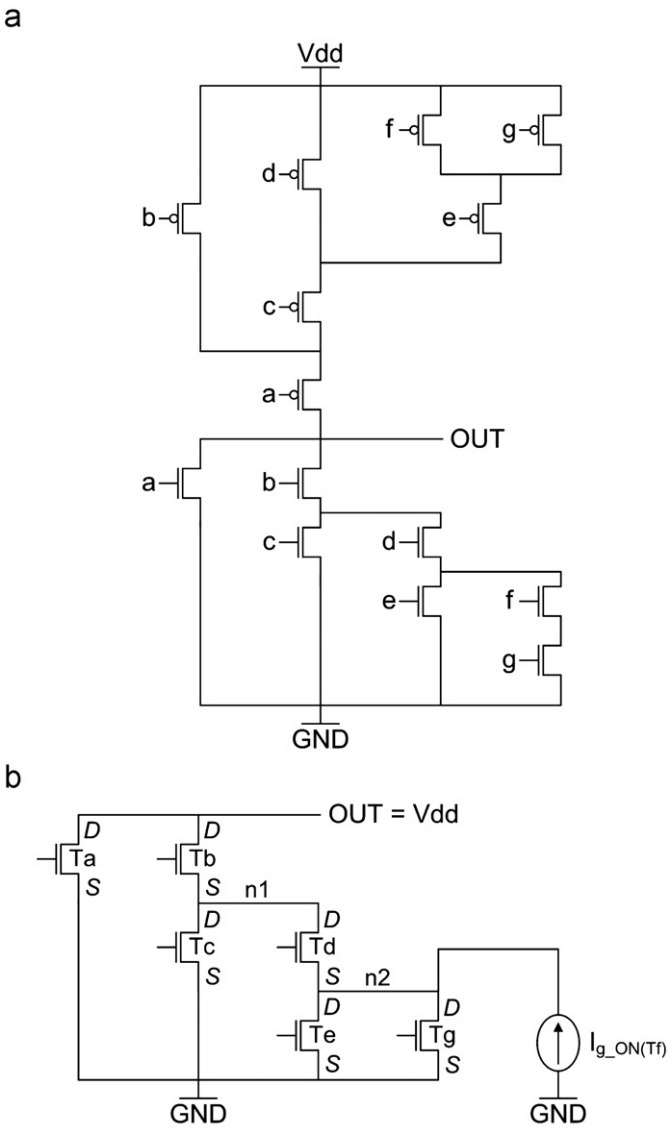

Fig. 7. Method illustration: (a) CMOS complex gates and (b) off-network representation for input vector [*a*, *b*, *c*, *d*, *e*, *f*, *g*]=[0,0,0,0,0,1,0].

Step 1: taking into account the input vector [*a*, *b*, *c*, *d*, *e*, *f*, *g*]=[0,0,0,0,0,1,0], the pull-down NMOS arrangement acts as the off-plane, while the pull-up PMOS plane is conducting.

Step 2: in the extraction of the off-network from the off-plane, the NMOS transistor with input '*f*' (*Tf*) is short-circuited and its gate oxide leakage contribution is associated to the respective node '*n*2' through a source current I_{g_ON}, as shown in Fig. 7b.

Step 3: the biasing is defined by taking into account the distance of the transistor nodes to the off-network terminals. The resulting biasing of the transistors is also depicted in Fig. 7b.

Step 4: the ordering to calculate the internal unknown node voltages is at first '*n*1' and then '*n*2'. The ordering is also defined according to the internal nodes distance to the output terminal.

Step 5: an initial drain-to-source voltage (V_{ds}) is computed to each transistor of the off-network in this step. When the node '*n*1' is evaluated, all other unknown voltages (node '*n*2') are considered at ground potential (GND), and Eq. (9) is used to find the V_{ds} of transistors *Tb*, *Tc*, and *Td*:

$$I_{s(Tb)} = I_{s(Tc)} + I_{s(Td)} + I_{g\_OFF(Tb)} + I_{g\_OFF(Tc)} + I_{g\_OFF(Td)} \quad (9)$$

where $I_{s(Ti)}$ and $I_{g\_OFF(Ti)}$ are, respectively, the subthreshold current and the off-device gate leakage current at each indicated

transistor *Tb*, *Tc*, and *Td*. The same notation is adopted in Eqs. (10) and (13). In those Equations, $I_{g\_ON(Ti)}$ is the on-device gate leakage current in '*i*' indexed transistors.

In the sequence, during the evaluation of node '*n*2', the V_{ds} of transistors *Te* and *Tg* are obtained by using Eq. (10):

$$I_{s(Td)} + I_{g\_ON(Tf)} = I_{s(Te)} + I_{s(Tg)} + I_{g\_OFF(Td)} + I_{g\_OFF(Te)} + I_{g\_OFF(Tg)} \quad (10)$$

Step 6: the initial drain-to-source voltages of each transistor are used to compute the unknown internal voltages in nodes '*n*1' and '*n*2' (V_{n1} and V_{n2}, respectively), by using Eqs. (11) and (12), respectively:

$$V_{n1} = V_{n2} + V_{ds(Tc)} \quad (11)$$

$$V_{n2} = V_{ds(Te)} = V_{ds(Tg)} \quad (12)$$

where V_{n1} and V_{n2} are the potentials in the nodes '*n*1' and '*n*2', respectively, and $V_{ds(Ti)}$ is the drain-to-source voltage in the respective transistors *Tc*, *Te*, and *Tg*.

The final V_{ds} value for each transistor are re-calculated based on '*n*1' and '*n*2' node potentials:

Step 7: the total leakage is the sum of all currents flowing from V_{dd} terminal or to GND terminal. Eq. (13) computes all leakage currents flowing to GND terminal:

$$I_{leakage} = I_{s(Ta)} + I_{s(Tc)} + I_{s(Te)} + I_{s(Tg)} + I_{g\_OFF(Ta)} + I_{g\_OFF(Tb)} + I_{g\_OFF(Tc)} + I_{g\_OFF(Td)} + I_{g\_OFF(Te)} + I_{g\_OFF(Tg)} \quad (13)$$

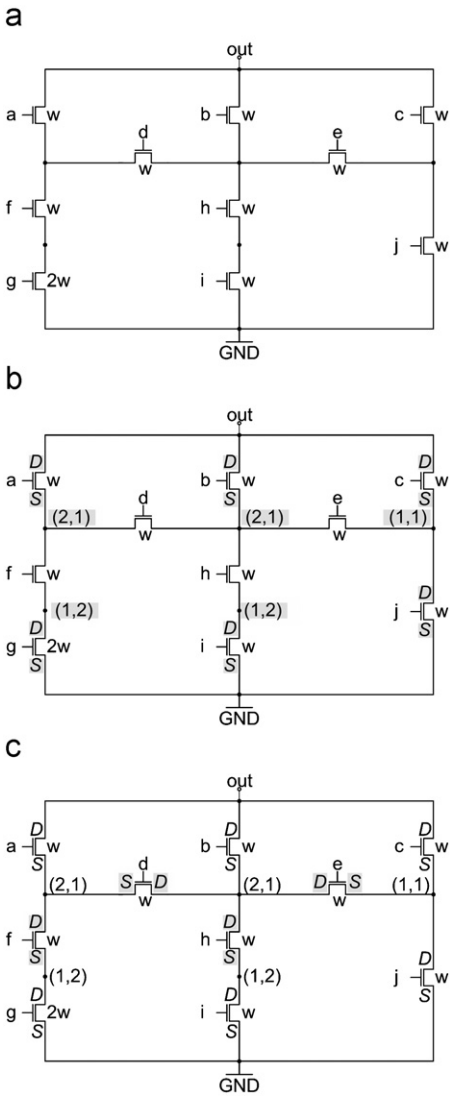

Fig. 8. Off-network representation: (a) originally extracted network; (b) assignments after steps 1, 2 and 3; (c) final transistor bias identification.

Non-series–parallel networks

The device DC polarity (biasing) identification, i.e. algorithm Step (3), in non-series–parallel off-network may not be a straightforward task as occuring in purely series–parallel transistor arrangements. The following procedure is used to set the bias condition of non-series–parallel arrangements, as the transistor network depicted in Fig. 8a.

(1) All transistor terminals directly connected to the power rail are assigned as source terminals, while the others are assigned as drain terminals;
(2) Similarly, all transistor terminals directly connected to the output are assigned as drain terminals, while the others are assigned as source terminals;
(3) Set the distance from each internal node to the off-network terminals (V_{dd} or GND, and output). These distances are indicated in Fig. 8b, being the numbers indicated in parentheses;
(4) For the transistors not connected directly to V_{dd} or GND terminals, the following rules are applied:
   (a) The transistor node closer to V_{dd} or GND is assigned as a source terminal, being the other assigned as a drain terminal;
   (b) If rule (IV.a) fails, the distances of both nodes to V_{dd} or GND are equal, then the distance to the gate output node is taken into account, being the closer node assigned as a drain terminal;
   (c) If rule (IV.b) fails, i.e. all node distances match among them, the transistor source terminal is defined as the one connected to the largest equivalent transistor calculated between the node and V_{dd} or GND, through series–parallel associations;
   (d) If rule (IV.c) fails, then the smaller equivalent transistor between the evaluated nodes and the output is applied, defining it as the source terminal;
   (e) In the case that all rules above fail, then such transistor biasing determination can be random, since the off-network presents a symmetric topology with respect to the transistor.

Although the proposed procedure above has been developed to define transistor biasing in bridge arrangements, it is also suitable for series–parallel networks. This way, it represents a general procedure for any kind of network topologies.

Fig. 8 exemplifies these steps to set the bias condition. For the off-network in Fig. 8a, all transistors connected directly to output and power rail have their terminals assigned. In addition, all nodes distances to GND and output terminals are defined in Fig. 8b. Furthermore, source and drain terminals for transistors controlled by signals '*e*', '*f*', and '*h*' are assigned using rule (IV.a) above. For the missing transistor, controlled by signal '*d*', the terminals are assigned according to the rule (IV.*c*). This is illustrated in Fig. 8c.

5. Experimental results

The proposed leakage estimation method was implemented, and their effectiveness has been evaluated. CAD tools were

written in Java platform, and experiments were carried out on a Sun Fire V890 Server with a CPU UltraSPARC IV 2.1 GHz, 8 GB of RAM, and SunOS 5.9. The obtained results were compared to HSPICE simulation data, considering 32 nm bulk CMOS PTM process [19].

5.1. Series–parallel logic gates

Logic gates, up to 6-inputs from GenLib_44-6 library [17], were analyzed for the complete truth table of the function that each gate implements. This set of cells is quite representative since it includes 93 different logic gate topologies with a maximum number of 4-stacked devices and 4 parallel devices in both pull-up PMOS and pull-down NMOS planes. The index '-6' indicates the logic depth, which represents the maximum alternating series and parallel transistor associations in the same logic plane.

Fig. 9 presents the average leakage current considering the application of all input vectors on each cell. The results demonstrate a good correlation between the estimated leakage values and the electrical simulation data. It can be verified by comparing the error magnitude to the leakage value that the difference is almost two decades. This results in an average error smaller than 7%.

5.2. Non-series–parallel CMOS gates

There are several works in the literature that explore the use of complex gates in circuit design. Kagaris et al. [18] present a methodology to design efficient non-series–parallel supergates. To the best knowledge of the authors, there are no previous works in literature that present an analytical solution to estimate the leakage current in such logic gates without using numerical solvers.

The leakage current for the CMOS gate depicted in Fig. 1b was estimated using the proposed method for all input vectors, and the results are compared to HSPICE data in Fig. 10. The difference between the estimated and simulated values is close to one decade ($\approx 10\%$) for low leakage currents. This can be observed on the left-side of the graph in Fig. 10. The model is much more accurate for the most significant leakage values, as it can be verified on the right-side of the graph in Fig. 10. Due to the higher accuracy for more significant values, the average error considering all possible input steady states is smaller than 2%.

5.3. Benchmark circuits

The method was also evaluated at circuit level, taking into account ISCAS85 benchmark circuits [21] mapped with ABC tool

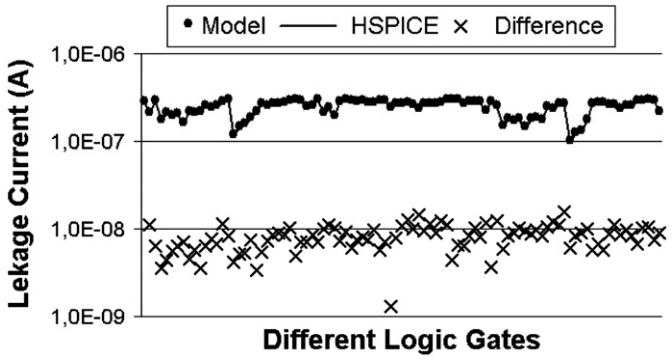

Fig. 9. Average leakage currents in 93 logic gates from GenLib_44-6 library [17], using a 32 nm bulk CMOS PTM process [19].

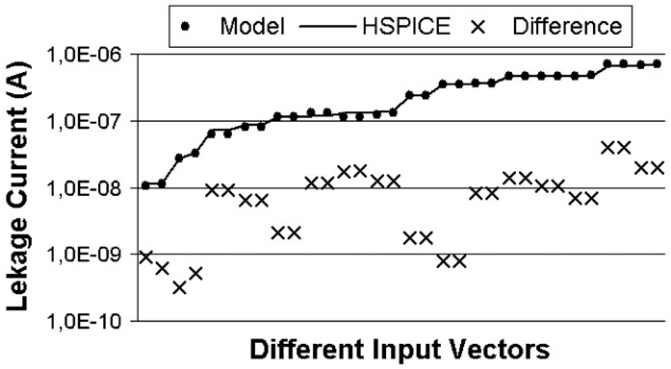

Fig. 10. Total leakage currents for all input vectors of the CMOS gate in Fig. 1b.

[22] to the GenLib_44-6 library [17]. The steady state of each internal cell in the circuit is pre-evaluated for a certain primary input vector. The leakage current in the cells is then calculated individually and summed for the total current estimation. Results obtained at circuit level for a hundred random vectors are shown in Table 3. The error is smaller than 2.6% for significantly faster runtimes, when comparing the model introduced herein to HSPICE simulations.

5.3.1. Analysis of method accuracy

Assumptions used in the proposed method leads to inaccuracies when compared to HSPICE results.

At first, the voltage drop observed in on-devices placed in the top of off-networks presents different values according to the number of stacked off-transistors. For instance, considering stacks with 2-, 3-, and 4-series NMOS transistors where only the top device (the one nearest to the output terminal) is turned on, as is illustrated in Fig. 11. The source-to-drain voltages over the top on-transistors are given in Table 4. Even for these small voltage differences, the impact in subthreshold and gate oxide leakages is quite significant.

Moreover, the assumption used in Section 4—Step (5), where "all unknown node voltages are initially considered at GND or V_{dd} potential, depending on the type of the off-plane, pull-down NMOS or pull-up PMOS, respectively", does not affect accuracy significantly. This happens because it is applied when there is a stack with three or more off-transistors. In this case, those internal nodes potential are close to GND or V_{dd} potential, e.g. around 30 mV for a three NMOS off-transistor stack in a 32 nm bulk CMOS PTM process and $V_{dd}=1.0$ V.

The choice done in Section 4—Step (6), i.e. "the voltage at this node is determined by the highest value obtained in the case of a NMOS off-network and by the lowest value in a PMOS network", introduces an inaccuracy, but it never underestimates the leakage current. For instance, considering the NMOS off-network used to exemplify the procedure in the new Section 4.1, the voltage value obtained by electrical simulation is $n1=128$ mV. In the case of choosing the lowest value ($n1=120$ mV), the leakage of the transistor *Tc* would be underestimated. Otherwise, when the highest value is chosen ($n1=138$ mV), the leakage of such transistor is overestimated. In this sense, the procedure proposed herein never underestimates the predicted total leakage currents.

Another source of error can be the interaction between the leakage current of different gates [23,24]. This interaction is known as *loading effect*. The estimation procedure described in Section 5.3 does not consider it. However, the small difference between the estimation and HSPICE values, observed in Table 3, leads to the affirmative of Mukhopadyay et al. in [23]: "depending on the input vector, the loading effect can either increase or

Table 3
Total leakage current estimation at circuit level.

Circuit	Number of transistors	Average leakage (μA)		Difference (%)		Run time (s)	
		Model	HSPICE	Average	Maximum	Model	HSPICE
C17	24	1.4	1.4	< 0.01	2.09	< 1	< 1
C8	694	29.1	29.1	< 0.01	1.12	13	41
C432	1210	50.3	49.7	1.26	1.98	39	125
C880	1776	89.9	89.2	0.69	1.27	45	223
C1908	2806	129.5	128.3	0.95	1.27	50	341
C499	3320	164.5	163.5	0.63	0.96	76	625
C1355	3396	165.3	164.3	0.59	0.96	90	550
C2670	3750	228.6	231.9	1.41	2.26	70	841
C3540	5758	325.4	327.7	0.69	1.72	123	2192
C7552	9840	529.0	541.8	2.36	2.59	412	5115
C6288	17,344	778.7	770.4	1.08	1.36	693	54,389

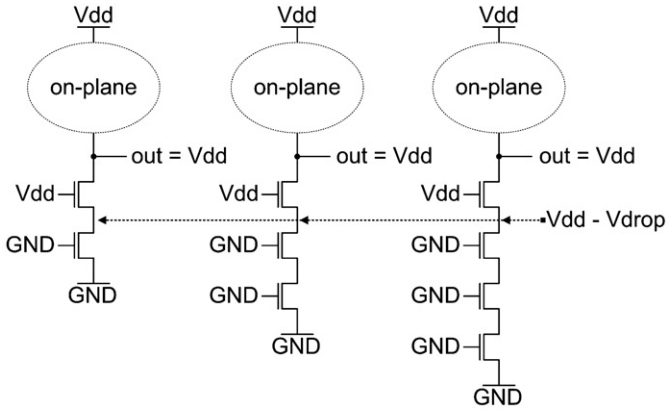

Fig. 11. On-devices on the top of stacks with different number of off-transistors.

Table 4
Voltage over the top on-transistors in NMOS transistors stacks depicted in Fig. 11.

Bulk CMOS PTM process (nm)	45	32	22
2-Stacked transistors (NAND2) (mV)	263	244	247
3-Stacked transistors (NAND3) (mV)	200	172	169
4-Stacked transistors (NAND4) (mV)	180	155	153

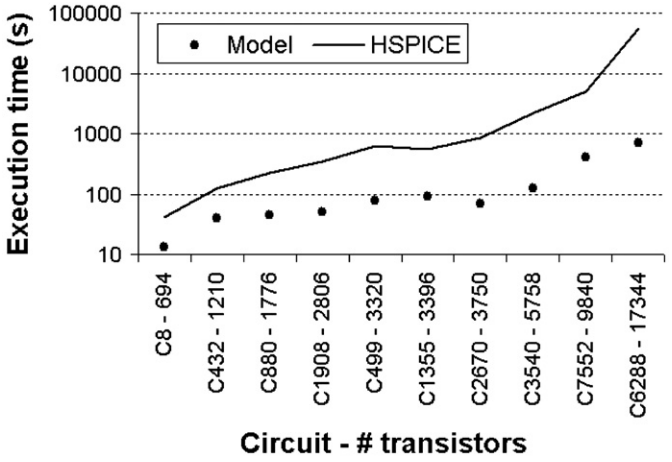

Fig. 12. Comparison of execution times.

decrease the leakage of a logic gate. Hence, in large circuits, the effect of loading on different logic gates may cancel out each other."

5.3.2. Execution time

The execution times, illustrated in Fig. 12, have been obtained by applying a hundred random vectors on the corresponding circuits. The proposed method is faster than electrical simulation, and the speed-up increases for larger circuits (in terms of transistor count). Note that the method has been implemented in Java programming language, while that could maybe improved by migrating it to C++ one, for instance. Nevertheless, the speed-up achieves 80 times for the largest evaluated circuit.

6. Conclusions

A new leakage current estimation method for CMOS circuits has been presented. Compared to other estimation techniques, this is an useful and efficient approach to treat any kind of transistor network configuration, including complex gates with more than two levels of series–parallel association as well as non-series–parallel arrangements. Subthreshold and gate oxide leakages are both considered in the analysis by interacting such leakage mechanisms. The proposed algorithm defines a sequential procedure, making assumptions which simplify the KCL resolution, while maintaining sufficient accuracy.

Acknowledgements

Research partially funded by Nangate Inc under a Nangate/UFRGS research agreement, by CNPq Brazilian funding agency, and by the European Community's Seventh Framework Programme under grant 248538-Synaptic.

References

[1] K. Roy, et al., Leakage current mechanisms and leakage reduction techniques in deep-submicron CMOS circuits, *Proceedings of IEEE* 91 (2) (2003) 305–327.
[2] 2007 International Technology Roadmap for Semiconductors (ITRS). [Online]. Available: http://public.itrs.net.
[3] E. Gusev, et al., Ultrathin high-K gate stacks for advanced CMOS devices, IEDM Technical Digest (2001) 451–454.
[4] A. Agarwal, et al., Leakage power analysis and reduction: models, estimation and tools, IEE Proceedings—Computers and Digital Techniques 152 (3) (2005) 353–368.
[5] S. Narendra, et al., Full-chip subthreshold leakage power prediction and reduction techniques for sub-0.18 um CMOS, IEEE Journal of Solid-State Circuits 39 (3) (2004) 501–510.
[6] R.X. Gu, M.I. Elmasry, Power distribution anaylsis and optimization of deep submicron CMOS digital circuits, IEEE Journal of Solid-State Circuits 31 (5) (1996) 707–713.
[7] Z. Chen, et al., Estimation of standby leakage power in CMOS circuits considering accurate modeling of transistor stacks, Proceedings of the

International Symposium on Low Power Electronics and Design (ISLPED), 1998, pp. 239–244.
[8] J.L. Rossello, J. Segura, Accurate modeling of leakage currents in nanometer CMOS technologies, Electronics Letters 41 (3) (2005) 122–124.
[9] H. Al-Hertani, D. Al-Khalili, C. Rozon, UDSM subthreshold leakage model for NMOS transistor stacks, Microelectronics Journal 39 (2008) 1809–1816.
[10] R.M. Rao, et al., Efficient techniques for gate leakage estimation, Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED), 2003, pp. 100–103.
[11] A. Ono, et al., A 100 nm node CMOS technology for practical SOC application requirement, Proceedings of the International Electron Devices Meeting, 2001, pp. 511–514.
[12] H. Rahman, C. Chakrabarti, A leakage estimation and reduction technique for scaled CMOS logic circuits considering cate leakage, Proceedings of the International Symposium on Circuits and Systems (ISCAS), vol. 2, 2004, pp. 297–300.
[13] D. Lee, D. Blaauw, D. Sylvester, Gate oxide leakage current analysis and reduction for VLSI circuits, IEEE Transactions on VLSI Systems 12 (2) (2004) 155–166.
[14] H. Al-Hertani, D. Al-Khalili, C. Rozon, Accurate total static leakage current sstimation in transistor stacks, Proceedings of the International Conference on Computer Systems and Applications (AICCSA), 2006, pp. 262–265.
[15] S. Yang, et al., Accurate stacking effect macro-modeling of leakage power in sub-100 nm circuits, Proceedings of the International Conference on VLSI Design, 2005, pp. 165–170.
[16] S. Mukhopadhyay, A. Raychowdhury, K. Roy, Accurate estimation of total leakage in nanometer-scale bulk CMOS circuits based on device geometry and doping profile, IEEE Transactions on Computer-Aided Design Integrated Circuits Systems 24 (2) (2005) 363–381.
[17] E.M. Sentovich, et al., SIS: a System for Sequential Circuit Synthesis, Electronic Res. Lab., Univ. California, Berkeley, CA, 1992 Technical Report. UCB/ERL M92/41.
[18] D. Kagaris, T. Haniotakis, A methodology for transistor-efficient supergate design, IEEE Transactions on VLSI Systems 15 (4) (2007) 488–492.
[19] W. Zhao, Y. Cao, New generation of predictive technology model for sub-45 nm early design exploration, IEEE Transactions on Electron Devices 53 (11) (2006) 2816–2823.
[20] B. Sheu, J.H. Shieh, M. Patil, BSIM: Berkeley short-channel IGFET model for MOS transistors, IEEE Journal of Solid-State Circuits SC-22 (1987) 558–566.
[21] D. Bryan, The ISCAS'85 Benchmark Circuits and Netlist Format, North-Carolina State University, 1985.
[22] Berkeley Logic Synthesis and Verification Group, ''ABC: A System for Sequential Synthesis and Verification'', Release 70930 [Online]. Avaliable: 〈http://www.eecs.berkeley.edu/~alanmi/abc/〉.
[23] S. Mukhopadhyay, S. Bhunia, K. Roy., Modeling and analysis of loading effect on leakage of nanoscaled bulk-CMOS logic circuits, IEEE Transactions on Computer-Aided Design Integrated Circuits Systems 25 (8) (2006) 1486–1495.
[24] A. Rastogi, W. Chen, S. Kundu., On estimating impact of loading effect on leakage current in sub-65 nm scaled CMOS circuits based on Newton-Raphson method, Proceedings of the Design Automation Conference (DAC) (2007) 712–715.